CLAIMS

What is claimed is:

1. A method of distributing workload in a workflow management system comprising the steps of:

a) calculating a load index for each engine of the workflow management system, wherein each load index reflects a workload of its associated engine, wherein the load index corresponds to an average activity execution delay; and

b) distributing workload across the plurality of engines in a load sensitive mode.

2. The method of claim 1 further comprising the steps of:

c) executing a test process to identify load index parameters including a single engine nominal activity execution delay (C) when no concurrent activities are executing and an activity execution latency factor (λ), wherein λ is a function of a number of concurrently executing activities.

3. The method of claim 2 wherein step a) further comprises the step of calculating the load index for each engine j as a total average activity execution delay $L(j) = C + \frac{1}{k}\sum_{i=1}^{k} N_i\lambda_i$, wherein k is a total number of activities completed within a pre-determined time period for engine j, wherein N_i is the number of other concurrently executing processes at the time activity i is executing, wherein λ_i is an execution latency rate for activity i.

4. The method of claim 2 wherein step a) further comprises the step of calculating the load index for each engine j as a relative average activity execution delay $L(j)=\frac{1}{k}\sum_{i=1}^{k} N_i\lambda_i$, wherein k is a total number of activities completed within a pre-determined time period for engine j, wherein N_i is the number of other concurrently executing activities at the time activity i is executing, wherein λ_i is an execution latency rate for activity i.

5. The method of claim 1 wherein step b) further comprises the step of re-directing incoming process requests to another engine.

6. The method of claim 1 wherein step b) further comprises the step of re-distributing queued processes to another engine.

7. The method of claim 1 wherein step b) further comprises the step of prioritizing a source engine for distributing workload from based on a maximum differential workload.

8. The method of claim 1 wherein step b) further comprises the step of identifying a target engine for distributing workload to based on a maximum differential workload.

9. A method of distributing workload in a workflow management system comprising the steps of:

a) calculating a load index for each engine of the workflow management system, wherein each load index reflects a workload of its associated engine;

b) operating in a load insensitive workload distribution mode for distributing processes until a maximum differential load index exceeds a pre-determined threshold; and

c) operating in a load sensitive workload distribution mode for distributing processes until all processes have completed execution once the maximum differential load index exceeds the pre-determined threshold.

10. The method of claim 9 wherein processes are round-robin distributed in the load insensitive workload distribution mode.

11. The method of claim 9 wherein step a) further comprises the step of calculating the load index for each engine j as a total average activity execution delay $L(j) = C + \frac{1}{k}\sum_{i=1}^{k} N_i\lambda_i$, wherein k is a total number of activities completed within a pre-determined time period for engine j, wherein N_i is the number of other concurrently executing processes at the time activity i is executing, wherein λ_i is an execution latency rate for activity i, wherein C is a single engine nominal activity execution delay when no concurrent activities are executing.

12. The method of claim 9 wherein step a) further comprises the step of calculating the load index for each engine j as a relative average activity execution delay $L(j) = \frac{1}{k}\sum_{i=1}^{k} N_i\lambda_i$, wherein k is a total number of activities completed within a pre-determined time period for engine j, wherein N_i is the number of other concurrently executing activities at the time activity i is executing, wherein λ_i is an execution latency rate for activity i.

13. The method of claim 9 wherein step c) further comprises the step of re-directing incoming process requests to another engine.

14. The method of claim 9 wherein step c) further comprises the step of re-distributing queued processes to another engine.

15. The method of claim 9 wherein step c) further comprises the step of prioritizing a source engine for distributing workload from based on a maximum differential workload.

16. The method of claim 9 wherein step c) further comprises the step of identifying a target engine for distributing workload to based on a maximum differential workload.

17. A method of distributing workload in a workflow management system comprising the steps of:

a) switching from a load insensitive mode to a load sensitive workload distribution mode for distributing processes when a maximum differential load index exceeds a first pre-determined threshold, T1; and

b) switching from the load sensitive mode to the load insensitive workload distribution mode for distributing processes when the maximum differential load index is less than a second pre-determined threshold, T2.

18. The method of claim 16 wherein T1=T2.

19. The method of claim 16 wherein T1 > T2.

20. The method of claim 17 wherein step a) further comprises the step of calculating a load index for each engine *j* as a total average activity execution delay $L(j) = C + \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein *k* is a total number of activities completed within a pre-determined time period for engine *j*, wherein N_i is the number of other concurrently executing processes at the time activity *i* is executing, wherein λ_i is an execution latency rate for activity *i*, wherein C is a single engine activity nominal execution delay when no concurrent activities are executing.

21. The method of claim 17 wherein step a) further comprises the step of calculating a load index for each engine *j* as a relative average activity execution delay $L(j) = \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein *k* is a total number of activities completed within a pre-determined time period for engine *j*, wherein N_i is the number of other concurrently executing activities at the time activity *i* is executing, wherein λ_i is an execution latency rate for activity *i*.